

# RANCANG BANGUN SISTEM *REAL TIME WATERMETER* BERBASIS *INTERNET OF THINGS (IOT)*

**AHS Budi\*, R Amshari dan B Mulyanti**

Departemen Pendidikan Teknik Elektro, Universitas Pendidikan Indonesia  
Jl. Dr. Setiabudhi No. 207, Bandung 40154, Indonesia.

\*agusheri@aol.com

**Abstrak:** Penelitian ini bertujuan untuk memperoleh sistem real time watermeter (RTWm) berbasis Internet of Things (IoT) yang dapat mengirimkan data konsumsi air ke website menggunakan sensor water flow, valve, dan ESP32. Metode yang digunakan dalam penelitian ini adalah metode eksperimen. Hasil penelitian menunjukkan bahwa sistem RTWm dengan rata-rata error relative sensor sebesar 2,39% telah. Dalam sistem watermeter ini, pemantauan penggunaan volume air dan harga yang harus dibayarkan dapat ditampilkan melalui website secara real time. Dengan teknologi IoT, sistem RTWm ini dapat memberikan umpan balik (feedback) jika pelanggan bermasalah dalam melakukan pembayaran.

**Kata Kunci:** watermeter, sensor water flow, valve, ESP32

**Abstract:** This study aims to obtain a real time watermeter (RTWm) system based on Internet of Things (IoT) which is able to send data of water consumption to the website using water flow sensors, valves, and ESP32. The method used in this research is the experimental method. The results showed that the RTWm system with a relative average error of 2.39% on sensor readings, has been successfully obtained. In this RTWm system, monitoring of water consumption and the cost to be paid can be displayed on the website in real time. In addition, by using IoT technology, this RTWm system can provide feedback when companies experience payment problems of customer.

**Keyword:** watermeter, sensor water flow, valve, ESP32

## 1. Pendahuluan

Air merupakan kebutuhan utama manusia yang pemanfaatannya semakin meningkat dari waktu ke waktu. Dengan semakin meningkatnya pertumbuhan industri, urbanisasi dan populasi, maka konsumsi air juga semakin meningkat [1]. Kelangkaan air dapat menyebabkan terganggunya kehidupan manusia dan sekitarnya [2].

Di Indonesia, pada umumnya air bersih dikelola oleh PDAM atau Perusahaan Daerah Air Minum, yaitu perusahaan milik Pemerintah Daerah yang bergerak pada distribusi air minum/air bersih bagi masyarakat [3]. Penggunaan air PDAM oleh konsumen, pada umumnya diukur dengan menggunakan meteran air atau *watermeter*, yang berfungsi untuk mengukur volume air yang digunakan dalam satuan m<sup>3</sup>. Bentuk fisik alat penunjuk pada *watermeter* dapat berupa digit angka, yang kemudian dicatat secara manual oleh petugas pemeriksa [4]. Untuk pengecekan jumlah penggunaan air oleh pihak PDAM dilakukan dengan cara mengirimkan petugas ke masing-masing pelanggan setiap bulan serta mencatatnya satu persatu. Cara tersebut kurang efektif dan efisien serta membutuhkan banyak tenaga dan biaya serta menghabiskan banyak waktu [5]. Di samping itu, pencatatan secara manual juga menimbulkan resiko yang tinggi akan terjadinya kesalahan dan ketidak-akuratan data [6].

Pada tahun 2001 hingga Agustus 2019, PDAM mengalami kerugian sebesar Rp 138.198.431.945 miliar. Kerugian tersebut diakibatkan oleh pelanggan yang menunggak atau tidak melakukan pembayaran penggunaan air [7].

Di negara lain, misalnya di Kerajaan Bahrain, sebelum tahun 2018 mereka masih menggunakan *watermeter* yang diperiksa secara manual setiap bulannya oleh petugas dari EWA (*Electricity and Water Authority*). Akan tetapi sejak tahun 2018, mereka telah mengganti semua divais dengan *watermeter* yang terhubung dengan internet. Dengan divais tersebut maka pemeriksaan penggunaan air menjadi lebih akurat dan tidak terjadi lagi kesalahan akibat dari kesalahan baca yang dilakukan oleh petugas [8].

Begitupun untuk sebagian besar tempat tinggal domestik di Australia, pembacaan *watermeter* dilakukan setiap tiga bulan. *Watermeter* yang digunakannya pun masih menggunakan *watermeter* konvensional. Akan tetapi, pada tahun 2018 Australia mulai menggunakan *Smart Water Metering* (SWM). SWM merupakan *watermeter* standar atau konvensional, dan pada dasarnya melakukan fungsi yang sama seperti *watermeter* standar atau konvensional. Namun, yang membedakannya adalah *watermeter* tersebut dapat membaca pengukuran penggunaan air dari jarak jauh [9].

Di Perancis beberapa otoritas membagi jaringan air mereka ke dalam sub-area distribusi air, dan memberikan pelanggan air dengan *smart watermeter*. *Smart Watermeter* tersebut dapat membaca penggunaan air dari jarak jauh serta pelanggan dapat memantau penggunaan air harian mereka melalui aplikasi yang telah disediakan. Serta diberikan peringatan (dikirimkan melalui SMS atau *-email*) untuk memberi tahu mereka ketika konsumsi air melebihi batas yang telah ditentukan sebelumnya, sehingga pelanggan dapat menghemat dalam penggunaan air [10].

IoT merupakan sebuah konsep jaringan yang memungkinkan membuat beberapa perangkat untuk saling berkomunikasi satu sama lain [11]. Pembacaan meter air yang sudah tersedia biasanya menggunakan GSM, GPRS, WI-FI, dll [12]. Namun penggunaan GSM misalnya, memiliki beberapa kelemahan seperti harus tersedia pulsa untuk dapat mengirim data serta *realtime* [6].

Untuk membantu mengatasi permasalahan yang dihadapi PDAM dan dengan melihat pengalaman di negara-negara lain, maka penelitian ini menawarkan solusi untuk PDAM, yaitu pembuatan sistem *watermeter* dengan teknologi pembacaan jarak jauh secara *realtime* dan sekaligus untuk mengendalikan *valve* menggunakan IoT.

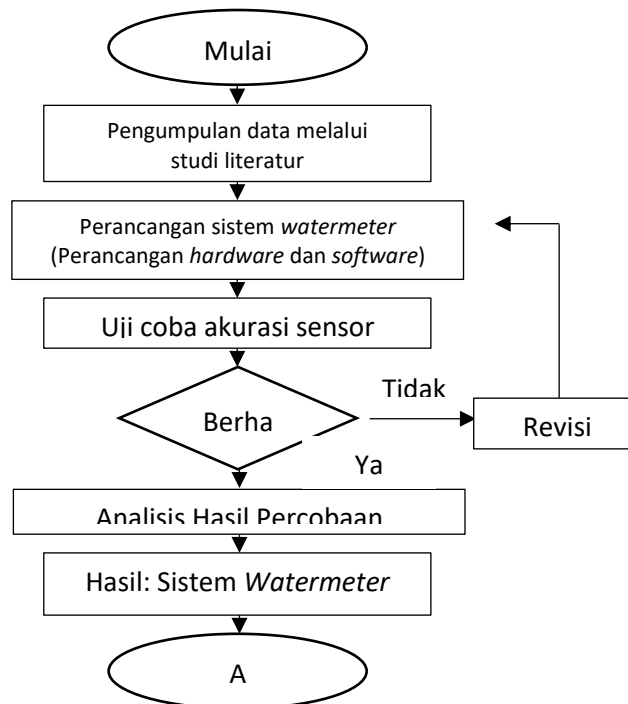
## 2. Metode Penelitian

Metode yang digunakan untuk melaksanakan penelitian ini adalah metode eksperimen. Metode eksperimen digunakan untuk merancang bangun sistem *watermeter* yang memiliki tingkat akurasi yang tinggi, sistem *real time watermeter* yang dapat melakukan pemantauan penggunaan air dan menghitung harga yang harus dibayar pelanggan, dan sistem *watermeter* berbasis IoT yang dapat melakukan umpan balik (*feedback*).

1. Perancangan sistem *watermeter*

Diagram alur untuk perancangan sistem *watermeter* yang memiliki tingkat akurasi yang tinggi seperti diperlihatkan pada Gambar 1.

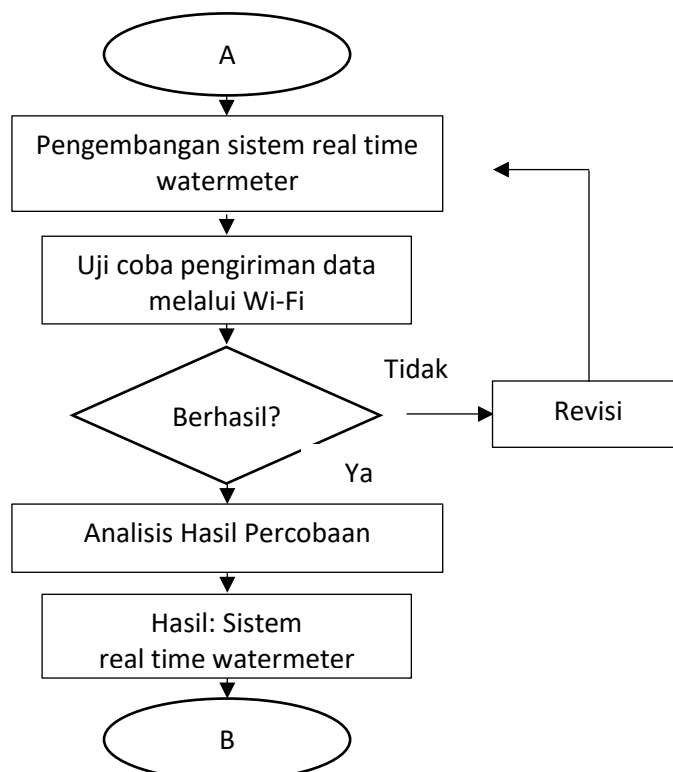
RANCANG BANGUN SISTEM *REAL TIME WATERMETER*  
BERBASIS *INTERNET OF THINGS (IOT)*



**Gambar 1.** Diagram alur untuk sistem *watermeter* yang andal

2. Pengembangan sistem *real time watermeter*

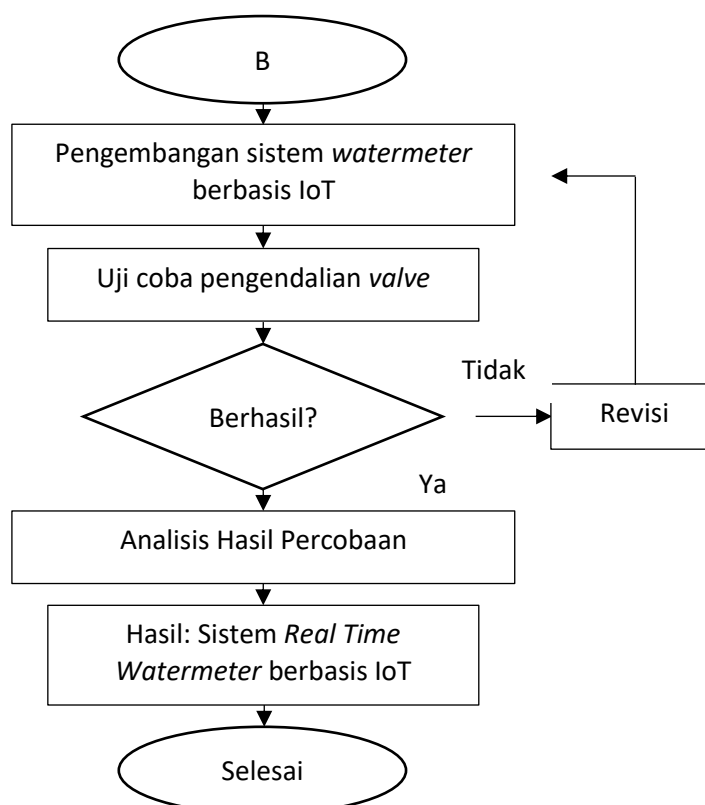
Diagram alur untuk sistem *real time watermeter* yang dapat melakukan pemantauan penggunaan air dan menghitung harga yang harus dibayar pelanggan diperlihatkan seperti pada Gambar 2.



**Gambar 2.** Diagram alur sistem *real time watermeter*

3. Pengembangan sistem *watermeter* berbasis IoT

Diagram alur untuk sistem *watermeter* berbasis IoT yang dapat melakukan umpan balik (*feedback*) diperlihatkan pada Gambar 3.



**Gambar 3.** Sistem *watermeter* berbasis IoT yang dapat melakukan umpan balik (*feedback*)

Tahap perancangan (tahap I) dilakukan untuk menghasilkan sistem *watermeter* yang memiliki tingkat akurasi yang baik. Pada tahap II dilakukan pengembangan sistem *real time watermeter* yang dapat melakukan pemantauan penggunaan volume air dan menghitung harga yang harus dibayar pelanggan. Tahap terakhir (tahap III) dilakukan pengembangan sistem *watermeter* berbasis IoT yang dapat melakukan umpan balik (*feedback*).

Pada tahap I dilakukan perancangan *hardware* dan *software* untuk menghasilkan sistem *watermeter* yang memiliki tingkat akurasi yang tinggi. Perancangan *hardware* dilakukan menyiapkan dan menyusun meteran air, sensor *water flow*, *valve*, mikrokontroler ESP32, modul relay 5V, dan baterai menjadi satu disambungkan dengan pipa dan pompa air untuk dibuat prototipenya. Kemudian diintegrasikan dengan mikrokontroler ESP32.

Untuk perangkat *software* dilakukan dengan membuat program yang dapat membaca volume air yang melewati meteran air dan sensor. Program tersebut dibuat menggunakan aplikasi Arduino IDE. Setelah perancangan *hardware* dan *software* selesai dibuat, maka tahap selanjutnya adalah melakukan uji coba akurasi sensor untuk menghasilnya sistem *watermeter* yang memiliki tingkat akurasi yang tinggi.

Selanjutnya tahap II dilakukan pengembangan sistem *watermeter* menjadi sistem *real time watermeter*. Sistem tersebut dapat dihasilkan dari pengembangan *software*. Program pembacaan volume air ditambahkan dengan program pengiriman data disatukan dan diintegrasikan oleh mikrokontroler ESP32. Sehingga menghasilkan sistem *real time watermeter* yang dapat memantau penggunaan volume air dan menghitung harga yang harus dibayarkan.

Pada tahap III dilakukan pengembangan sistem *watermeter* berbasis IoT yang dapat melakukan umpan balik (*feedback*). Sistem tersebut dapat melakukan kontrol *valve* dengan menggunakan IoT.

### 3. Hasil dan Pembahasan

Hasil dari penelitian ini adalah menghasilkan sistem *real time watermeter* berbasis IoT.

#### 3.1 Perancangan Sistem *Watermeter*

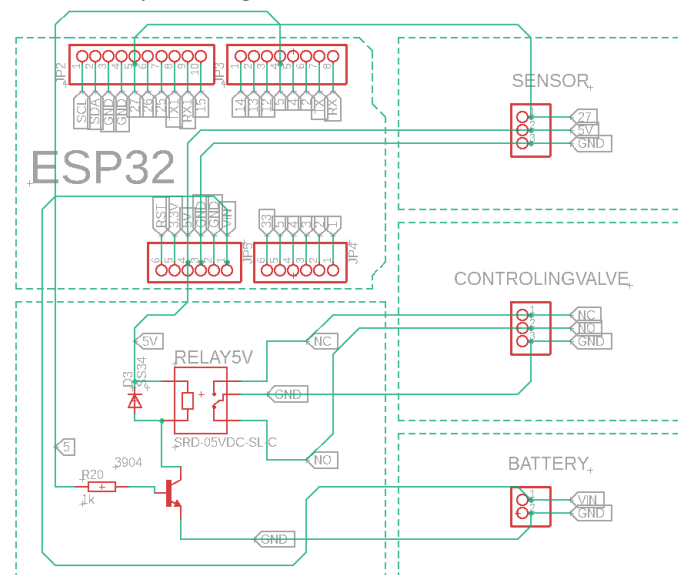
Perancangan sistem *watermeter* dilakukan dengan merancang *hardware* dan *software*.

##### 3.1.1 Perancangan *hardware* sistem *watermeter*

Tahap awal dalam membuat rancang bangun sistem *real time watermeter* berbasis IoT adalah dengan membuat *schematic* dan menyiapkan *hardware* yang dibutuhkan. *Hardware* yang dibutuhkan adalah sebagai berikut:

1. *Watermeter* atau meteran air
2. Sensor *water flow*
3. *Valve*
4. Mikrokontroler ESP32
5. *Relay*
6. Baterai
7. Kabel USB
8. Pompa air
9. Pipa
10. Akurium

*Schematic* prototipe *watermeter* merupakan rancangan komunikasi antara mikrokontroler, sensor, relay, dan baterai. Gambar 4. merupakan *schematic* prototipe *watermeter* yang dapat melakukan pemantauan volume air serta dapat mengendalikan *valve*.



**Gambar 4.** *Schematic* sistem *watermeter*

Pada Gambar 4 ESP32 merupakan mikrokontroler utama yang dapat mengendalikan relay, sensor, *valve*, dan baterai. Sensor *water flow* dihubungkan dengan pin digital 27 ESP32, dihubungkan dengan tegangan 5V dan juga GND atau *Ground*. *Valve* dihubungkan dengan relay yang telah terhubung dengan pin digital 5 ESP32. Ketika relay diberi perintah 1, relay akan membuka *valve* dan ketika relay diberi perintah 0, relay akan menutup *valve*. Serta Baterai dihubungkan ke pin VIN dan juga GND ESP32.

*Hardware* yang telah disiapkan disusun menjadi satu bagian yang utuh yang ditunjukkan pada Gambar 5. Perancangan dilakukan dengan Menyusun pompa air dihubungkan dengan pipa, lalu

dihubungkan dengan *valve*, setelah itu dihubungkan dengan meteran air, dan terakhir dihubungkan dengan sensor yang telah terhubung ke sistem perangkat keras yang telah dibuat.



**Gambar 5.** Perancangan sistem *watermeter*

Rancangan sistem perangkat keras seperti pada Gambar 6 merupakan mikrokontroler ESP32 yang diberi *supply* tegangan 5V melalui *micro USB* dihubungkan dengan modul *relay* sebagai pengendali *valve*, baterai sebagai *supply* cadangan jika ESP32 tidak ada *supply* tegangan 5V, dan dihubungkan dengan sensor *water flow* melalui *pin* digital 27 ESP32 serta bekerja pada tegangan 5V. Fungsi sensor *water flow* ini adalah untuk membaca volume air yang telah melewati meteran air serta hasil data yang dibaca sensor akan dikirimkan ke platform ANTARES oleh ESP32 dengan menggunakan konektivitas Wi-Fi. Sistem perangkat keras inilah yang akan mengontrol semua sistem *real time watermeter* ini secara keseluruhan.



**Gambar 6.** Sistem perangkat keras prototipe *watermeter*

Modul relay dapat melakukan pengendalian *valve* ketika program *relay* diberi perintah 1 maka *relay* akan membuat *valve* terbuka sehingga air dapat melewati meteran air, dan Ketika *relay* diberi perintah 0 maka *relay* akan membuat *valve* tertutup sehingga air tidak dapat melewati meteran air.

### 3.1.2 Perancangan *software* sistem *watermeter*

Pada tahap perancangan *software* dibutuhkan *software* Arduino IDE. *Software* Arduino IDE berfungsi sebagai *uploader* program yang akan dibuat ke dalam mikrokontroler ES32. Tahapannya, pertama dilakukan *uploading* program (*coding*) untuk membaca volume air yang melewati meteran air dan sensor. Untuk *coding* yang dimaksud terdapat dalam Gambar 7. Pada *coding*-an di Gambar 7 terdapat *calibrationfactor* yang akan mempengaruhi dari nilai bacaan sensor. Nilai tersebut didapat dari *library* sensor.

## RANCANG BANGUN SISTEM *REAL TIME WATERMETER* BERBASIS *INTERNET OF THINGS (IOT)*

```
Volume | Arduino 1.8.12
File Edit Sketch Tools Help
Volume$
#define SENSOR 27

long currentMillis = 0;
long previousMillis = 0;
int interval = 1000;
boolean ledState = LOW;
float calibrationFactor = 6.2;
volatile byte pulseCount;
byte pulseSec = 0;
float flowRate;
unsigned int flowMillilitres;
unsigned long totalMillilitres;

void IRAM_ATTR pulseCounter()
{
  pulseCount++;
}

void setup()
{
  Serial.begin(115200);

  // pinMode(LED_BUILTIN, OUTPUT);
  pinMode(SENSOR, INPUT_PULLUP);

  pulseCount = 0;
  flowRate = 0.0;
  flowMillilitres = 0;
  totalMillilitres = 0;
  previousMillis = 0;

  attachInterrupt(digitalPinToInterrupt(SENSOR), pulseCounter, FALLING);
}

void loop()
{
  currentMillis = millis();
  if (currentMillis - previousMillis > interval) {

    pulseSec = pulseCount;
    pulseCount = 0;

    // Because this loop may not complete in exactly 1 second intervals we calculate
    // the number of milliseconds that have passed since the last execution and use
    // that to scale the output. We also apply the calibrationFactor to scale the output
    // based on the number of pulses per second per units of measure (litres/minute in
    // this case) coming from the sensor.
    flowRate = ((1000.0 / (millis() - previousMillis)) * pulseSec) / calibrationFactor;
    previousMillis = millis();

    // Divide the flow rate in litres/minute by 60 to determine how many litres have
    // passed through the sensor in this 1 second interval, then multiply by 1000 to
    // convert to millilitres.
    flowMillilitres = (flowRate / 60) * 1000;

    // Add the millilitres passed in this second to the cumulative total
    totalMillilitres += flowMillilitres;

    // Print the flow rate for this second in litres / minute
    Serial.print("Flow rate: ");
    Serial.print(int(flowRate)); // Print the integer part of the variable
    Serial.print("L/min");
    Serial.print("\t"); // Print tab space

    // Print the cumulative total of litres flowed since starting
    Serial.print("Volume Air: ");
    Serial.print(totalMillilitres);
    Serial.print("mL / ");
    Serial.print(totalMillilitres / 1000);
    Serial.println("L");
  }
}
```

**Gambar 7.** Coding mengukur volume air

*Coding* sensor *water flow* bekerja untuk membaca debit air yang melewati sensor dan merubahnya menjadi volume. *Calibrationfactor* merupakan faktor pembagi pada *coding* sensor *water flow*. Apabila nilai *calibrationfactor* diubah menjadi lebih besar, maka volume air yang terbaca oleh sensor akan kurang dari volume yang seharusnya dan apabila nilai *calibrationfactor* diubah menjadi lebih kecil, maka volume air yang terbaca oleh sensor akan lebih besar dari volume yang seharusnya.

### 3.1.3 Pengujian akurasi sensor

Pengujian akurasi sensor bertujuan untuk mengetahui akurasi dari *sensor water flow* yang telah terhubung ke meteran air. Percobaan dilakukan dengan cara membandingkan hasil air yang telah terukur dan yang terbaca oleh sensor seperti contoh pada Gambar 8 yang membandingkan hasil bacaan meteran air dari Volume awal 962,45 m<sup>3</sup> sampai 962,46 m<sup>3</sup>. Hasil tersebut menunjukkan bahwa air telah mengalir melewati meteran air tersebut sebanyak 10 L atau 10000 mL.



**Gambar 8.** Meteran air sebelum dan sesudah dilewati air

Untuk menguji akurasi sensor, maka dilakukan pengujian dengan membaca 3 sampel pengujian dan masing-masing sampel diuji sebagai 20 kali. Untuk hasil pengujiannya dapat dilihat pada Tabel 1, Tabel 2 dan Tabel 3 seperti dibawah ini.

**Tabel 1.** Pengujian akurasi sensor pada volume air 10 L

No	Volume air yang diukur (mL)	Meteran air sebelum pengujian (m <sup>3</sup> )	Meteran air setelah pengujian (m <sup>3</sup> )	Volume air yang dibaca sensor (mL)	Selisih (mL)	(Selisih) <sup>2</sup>
1	10000	962,42	962,43	10237	237	56169
2	10000	962,43	962,44	10045	45	2025
3	10000	962,44	962,45	10602	602	362404
4	10000	962,45	962,46	10499	499	249001
5	10000	962,46	962,47	10204	204	41616
6	10000	962,47	962,48	10502	502	252004
7	10000	962,48	962,49	10089	89	7921
8	10000	962,49	962,50	10284	284	80656
9	10000	962,50	962,51	10104	104	10816
10	10000	962,51	962,52	10130	130	16900
11	10000	963,20	963,21	10299	299	89401
12	10000	963,28	963,29	10425	425	180625
13	10000	963,29	963,30	10105	105	11025
14	10000	963,30	963,31	10328	328	107584
15	10000	963,31	963,32	10214	214	45796
16	10000	963,32	963,33	10654	654	427716
17	10000	963,33	963,34	9975	25	625
18	10000	963,34	963,35	10103	103	10609
19	10000	963,35	963,36	10250	250	62500
20	10000	963,36	963,37	10237	237	56169
Rata-rata volume hasil pembacaan				10264,3	Rata-rata error RMS	321,83



Perhitungan nilai rata-rata *error* dan akurasi pada pengujian volume air 10 L:

Rata-rata volume hasil pembacaan

$$\bar{S} = \frac{\sum_{k=1}^n S_k}{n}$$

Rata-rata *error* RMS pada volume

$$\Delta S_{rms} = \sqrt{\frac{\sum_{k=1}^n (\Delta S_k)^2}{n}}$$

Hasil Pembacaan

$$S = \bar{S} \pm \Delta S_{rms}$$

$$S = 10264,3 \pm 321,83 \text{ mL}$$

*Error relative*

$$e_r = \frac{\Delta S_{rms}}{\bar{S}} \times 100\%$$

$$e_r = \frac{321,83}{10264,3} \times 100\%$$

$$e_r = 3,14\%$$

Hasil pengujian akurasi sensor pada volume 20 Liter dapat dilihat dalam table 2 seperti disajikan dibawah ini.

**Tabel 2.** Pengujian akurasi sensor pada volume air 20 L

No	Volume air yang diukur (mL)	Meteran air sebelum pengujian (m <sup>3</sup> )	Meteran air setelah pengujian (m <sup>3</sup> )	Volume air yang dibaca sensor (mL)	Selisih (mL)	(Selisih) <sup>2</sup>
1	20000	962,42	962,43	20011	11	121
2	20000	962,43	962,44	20372	372	138384
3	20000	962,44	962,45	20810	810	656100
4	20000	962,45	962,46	20494	494	244036
5	20000	962,46	962,47	20344	344	118336
6	20000	962,47	962,48	20006	6	36
7	20000	962,48	962,49	19244	756	571536
8	20000	962,49	962,50	19460	540	291600
9	20000	962,50	962,51	19957	43	1849
10	20000	962,51	962,52	20026	26	676
11	20000	963,20	963,21	21404	1404	1971216
12	20000	963,41	963,43	20354	354	125316
13	20000	963,45	963,47	20061	61	3721
14	20000	963,47	963,49	20106	106	11236
15	20000	963,49	963,51	20221	221	48841
16	20000	963,51	963,53	20821	821	674041
17	20000	963,53	963,55	20039	39	1521
18	20000	963,55	963,57	20096	96	9216
19	20000	963,57	963,59	20257	257	66049
20	20000	963,59	963,61	20497	497	247009
Rata-rata volume hasil pembacaan				20229	Rata-rata <i>error</i> RMS	508,96

Perhitungan nilai rata-rata *error* dan akurasi pada pengujian volume air 20 L:

Rata-rata volume hasil pembacaan

$$\bar{S} = \frac{\sum_{k=1}^n S_k}{n}$$

Rata-rata *error* RMS pada volume

$$\Delta S_{rms} = \sqrt{\frac{\sum_{k=1}^n (\Delta S_k)^2}{n}}$$

Hasil Pembacaan

$$S = \bar{S} \pm \Delta S_{rms}$$

$$S = 20229 \pm 508,96 \text{ mL}$$

*Error relative*

$$e_r = \frac{\Delta S_{rms}}{\bar{S}} \times 100\%$$

$$e_r = \frac{508,96}{20229} \times 100\%$$

$$e_r = 2.52\%$$

Sedangkan hasil pengujian akurasi sensor pada volume 50 Liter dapat dilihat dalam tabel 3 seperti disajikan dibawah ini.

**Tabel 3.** Pengujian akurasi sensor pada volume air 50 L

No	Volume air yang diukur (mL)	Meteran air sebelum pengujian (m <sup>3</sup> )	Meteran air setelah pengujian (m <sup>3</sup> )	Volume air yang dibaca sensor (mL)	Selisih (mL)	(Selisih) <sup>2</sup> (mL)
1	50000	963,75	932,80	51700	1700	2890000
2	50000	962,43	963,85	50487	487	237169
3	50000	963,85	963,90	50062	62	3844
4	50000	963,90	963,95	51508	1508	2274064
5	50000	963,95	964,00	50536	536	287296
6	50000	964,00	964,05	50766	766	586756
7	50000	964,05	964,10	50031	31	961
8	50000	964,13	964,18	49796	204	41616
9	50000	964,18	964,23	50573	573	328329
10	50000	964,23	964,28	50391	391	152881
11	50000	964,30	964,35	50598	598	357604
12	50000	964,37	964,42	50039	39	1521
13	50000	964,42	964,47	50072	72	5184
14	50000	964,53	964,58	50280	280	78400
15	50000	964,58	964,63	50422	422	178084
16	50000	964,64	964,69	50870	870	756900
17	50000	964,75	964,80	50959	959	919681
18	50000	944,80	964,85	50318	318	101124
19	50000	964,85	964,90	51456	1456	2119936
20	50000	964,90	964,95	50278	278	77284

Rata-rata volume hasil pembacaan	50557,1	Rata-rata <i>error</i> RMS	754,94
----------------------------------	---------	----------------------------	--------

Perhitungan nilai rata-rata *error* dan akurasi pada pengujian volume air 50 L:

Rata-rata volume hasil pembacaan

$$\bar{S} = \frac{\sum_{k=1}^n S_k}{n}$$

Rata-rata *error* RMS pada volume

$$\Delta S_{rms} = \sqrt{\frac{\sum_{k=1}^n (\Delta S_k)^2}{n}}$$

Hasil Pembacaan

$$S = \bar{S} \pm \Delta S_{rms}$$

$$S = 50557,1 \pm 754,94 \text{ mL}$$

*Error relative*

$$e_r = \frac{\Delta S_{rms}}{\bar{S}} \times 100\%$$

$$e_r = \frac{754,94}{50557,1} \times 100\%$$

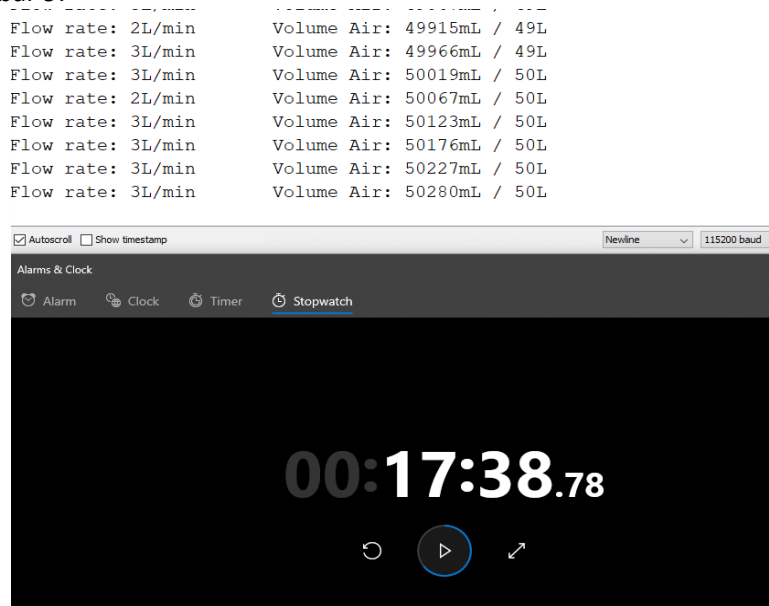
$$e_r = 1,5\%$$

Pada pengujian pertama mengukur air dengan volume 10L sebanyak 20 kali percobaan. Pengujian tersebut mendapatkan *error relative* sebesar 3,14%. Pada pengujian kedua dengan volume 20L mendapatkan *error relative* sebesar 2,52%. Serta pada pengujian ketiga dengan volume air 50L mendapatkan *error relative* sebesar 1,5%.

Dari ketiga percobaan yang dilakukan dapat dilihat hasilnya bahwa sensor tersebut memiliki akurasi 98.12% dan *error* 1.88% dalam membaca volume air. Dari ketiga sampel data yang diambil serta dengan 20 kali percobaan menghasilkan data yang akurat dalam membaca volume air yang melewati meteran air. Sensor *water flow* dapat membaca volume air dengan persamaan sebagai berikut:

$$\text{Volume Air (V)} = \text{Debit Air} \times \text{Waktu}$$

Volume air didapat dari hasil kali antara debit air dan juga waktu. Untuk membuktikannya dapat dilihat pada Gambar 9.



Gambar 9. Hubungan debit air dengan volume

Dari Gambar 9 dapat dilihat *flow rate* atau debit air sebesar 2-3L/menit. Dengan melihat hasil data tersebut, debit air sekitar 3L/menit.

$$\begin{aligned} \text{Volume Air (V)} &= 3\text{L/menit} \times 17.38 \\ \text{Volume Air (V)} &= 52140\text{mL} \end{aligned}$$

Volume air yang dihitung dengan yang terbaca oleh sensor berbeda 1860mL, itu semua terjadi karena debit air yang melewati sensor berubah ubah tergantung kondisi pompa air dan kondisi air yang tidak bercampur dengan gelembung udara. Jika air tercampur gelembung udara maka bacaan sensor dengan kondisi air yang terbaca akan berbeda, karena sensor *water flow* tidak bisa membedakan air yang memiliki banyak gelembung dengan air yang tanpa gelembung. Jika air tersebut melewati sensor *flow meter* maka tetap akan membaca sejumlah air yang melewati sensor tersebut meskipun air tersebut memiliki banyak gelembung udara

Dengan digunakannya sensor *water flow* dapat menjadi alternatif terbaik untuk membuat prototipe *watermeter* dengan harga terjangkau dan akurasi yang baik. Dengan meteran air yang telah disediakan oleh PDAM dapat dihubungkan dengan sistem perangkat keras yang telah dibuat dan dihubungkan dengan sensor *water flow* dapat membuat meteran air PDAM dapat dipantau dari jarak jauh serta petugas PDAM tidak perlu melakukan pengecekan volume air yang telah melewati meteran air tersebut

## 3.2 Pengembangan Sistem *Real Time Watermeter*

Sistem *real time watermeter* merupakan sistem *watermeter* yang dapat memantau penggunaan volume air dan harga yang harus dibayarkan secara *real time* serta dapat ditampilkan pada *website*. Perbedaan pada sistem ini terdapat pada pengembangan *software*.

### 3.2.1 Pengembangan *software*

Program (*coding*) yang digunakan untuk menghasilkan sistem *realtime watermeter* adalah dengan menggabungkan *coding* membaca volume air dan *coding* AntaresStoreData. *Coding* AntaresStoreData merupakan *coding*-an yang berfungsi untuk mengirimkan data sensor yang di baca oleh mikrokontroler ESP32 ke platform ANTARES yang nantinya akan ditampilkan pada *website*. Gambar 10. Menunjukkan *coding* untuk membuat sistem *real time watermeter* yang dapat mengirim data ke platform dan juga ke *website*.



```
Skripsiwatermeterku | Arduino 1.8.12
File Edit Sketch Tools Help
Skripsiwatermeterku $
#include <AntaresESP32HTTP.h>
#include <ArduinoJson.h>

#define ACCESSKEY "c016e5fe10d47c7f:5ebb0f0636ae2936"
#define WIFISSID "Cosmic.id"
#define PASSWORD "cosmic12345"

#define projectName "Ridwan_Watermeter"
#define deviceNameStatus "Watermeter"
#define deviceNameControl "Controlvalve"

AntaresESP32HTTP antares(ACCESSKEY);

#define SENSOR 27
#define VALVE 5

#define OPEN 0
#define CLOSE 1

//int counter = 0;
long currentMillis = 0;
long previousMillis = 0;
int interval = 1000;
boolean ledState = LOW;
float calibrationFactor = 6.2;
volatile byte pulseCount;
byte pulse1Sec = 0;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;
```

**Gambar 10.** Coding prototipe real time watermeter

Pada Gambar 10 terdapat beberapa hal penting yang harus dipenuhi, diantaranya adalah *Accesskey*, *wifissid*, *password*, *projectName*, *deviceNameStatus*, *deviceNameControl*. Itu semua harus dipenuhi agar prototipe yang dibuat dapat mengirimkan data ke platform ANTARES.

Fungsi dari *Accesskey* adalah sebagai kunci dasar untuk mengakses akun ANTARES yang dimiliki. *Accesskey* dari setiap akun akan berbeda-beda. Fungsi *wifissid* dan *password* merupakan *sketch* untuk mengakses Wi-Fi. Lalu fungsi dari *projectName*, *deviceNameStatus*, *deviceNameControl* merupakan *sketch* untuk menunjukkan alamat data sensor yang akan dikirim.

### 3.2.2 Pengujian sistem *real time watermeter*

Pengujian sistem *real time watermeter* dilakukan dengan cara mengirimkan data sensor berupa volume air ke *platform* ANTARES dan ditampilkan di *website*. Hasil pengiriman data sensor ke *platform* ANTARES ditunjukkan pada Gambar 11.



Hasil dari pengujian rancang bangun sistem *real time watermeter* adalah data bacaan sensor dapat ditampilkan pada platform ANTARES dan *website* secara *real time*. Yang dimaksud *real time watermeter* adalah air yang melewati meteran air dapat terus dipantau pada *website* selama prototipe tetap terkoneksi dengan *Wi-Fi*. Namun, apabila prototipe tidak terkoneksi dengan *Wi-Fi* maka data hasil bacaan tidak bisa ditampilkan pada *website*. Akan tetapi, prototipe masih tetap melakukan membaca aliran air yang melewati sensor. Ketika prototipe terkoneksi lagi dengan *Wi-Fi* maka data terakhir yang dibaca oleh sensor akan dikirimkan kembali ke *platform* dan *website* serta terus melakukan pengiriman data kembali secara *real time*. Dengan sistem seperti itu, maka konsumen pengguna meteran air dapat terus memantau berapa air yang telah dipakai dan juga besaran harga yang harus dibayar.

### 3.3 Pengembangan Sistem *Watermeter* berbasis IoT

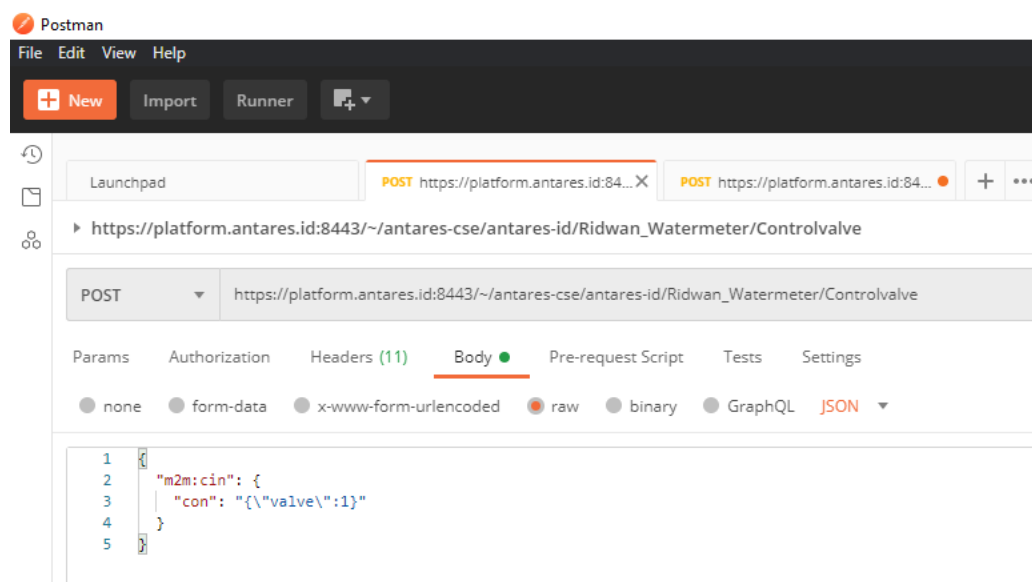
Hasil akhir dari penelitian ini adalah terciptanya sebuah prototipe *sistem real time watermeter* berbasis *Internet of Things (IoT)* yang dapat melakukan pemantauan penggunaan volume air dan dapat melakukan *feedback*.

#### 3.3.1 Pengujian *feedback* pada sistem *watermeter* berbasis IoT

Pada pengujian *feedback* atau pengujian *valve* dilakukan dengan menghubungkan *valve* dan mikrokontroler ESP32. Relay tersebut yang akan mengontrol *valve* untuk dapat berfungsi membuka dan menutup. Pengujiannya dilakukan 2 kali pengujian. Pengujian pertama membuka dan menutup *valve* menggunakan aplikasi Postman dan pengujian kedua membuka dan menutup *valve* menggunakan *website*.

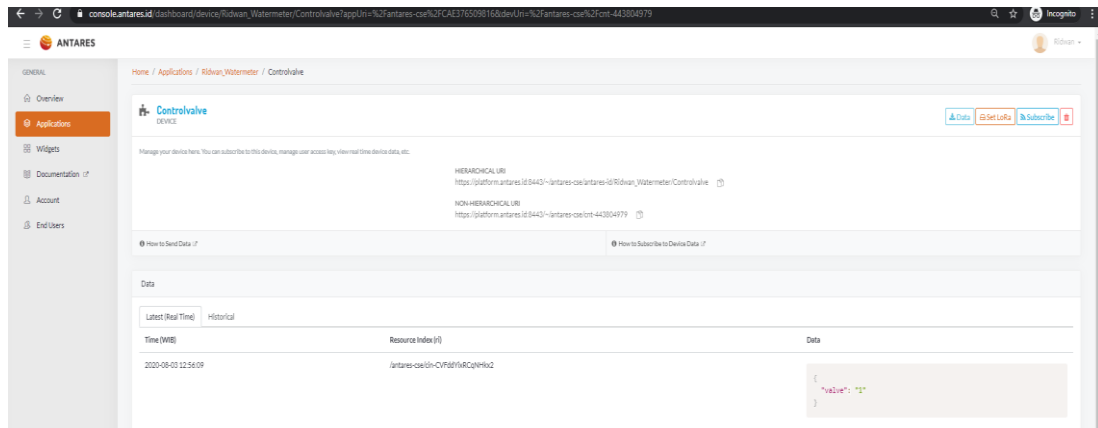
##### 1. Pengujian menggunakan aplikasi Postman

Pada aplikasi Postman dilakukan perintah POST untuk melakukan POST data ke platform ANTARES. Perintah POST ditunjukkan pada Gambar 13. Pada Gambar 13 dapat dilihat *coding* “{\valve\":1}”, yang memiliki arti bahwa *valve* diberikan perintah 1 yang akan membuat *valve* terbuka.



**Gambar 13.** Tampilan POST pada aplikasi Postman

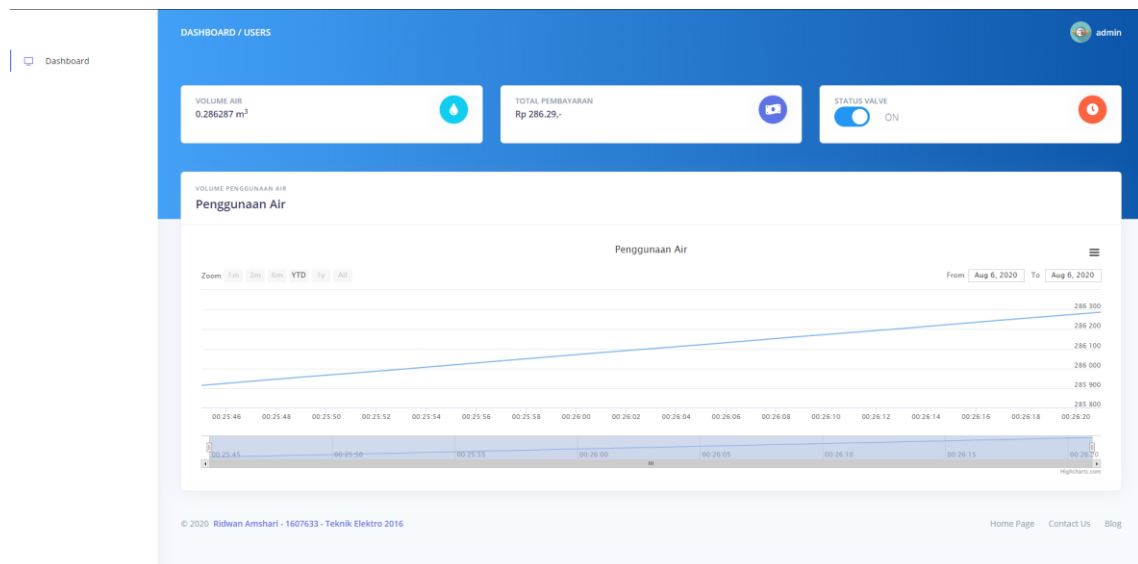
Ketika perintah POST tersebut dikirim maka data akan masuk ke ANTARES seperti pada Gambar 14.



**Gambar 14.** Data yang di-POST oleh Postman masuk ke platform ANTARES

2. Pengujian menggunakan *website*

Pengujian *valve* pada *website* dilakukan dengan cara melakukan klik pada menu “STATUS VALVE” yang terlihat pada Gambar 15.



**Gambar 15.** Tampilan *website* untuk *feedback valve*

Ketika *valve* statusnya ON atau sama dengan 1 maka *valve* akan terbuka dan air dapat melewati sistem *watermeter*. Tetapi, jika diberi perintah 0 maka *valve* akan menutup dan air tidak dapat melewati sistem *watermeter*. Untuk hasilnya seperti terlihat pada Gambar 16 Ketika *valve* terbuka.



**Gambar 16.** Kondisi *valve* Ketika terbuka



Ketika air mengalir maka sensor dapat membaca volume air yang telah melewati sistem *watermeter* serta dapat melakukan pengiriman data volume air ke *website* untuk dapat ditampilkan berapa besaran volume yang telah terbaca oleh sensor.

Sedangkan pada Gambar 17 menunjukkan valve dalam kondisi tertutup Namun, air tidak mengalir melewati sistem *watermeter* sehingga sistem *watermeter* tidak dapat membaca volume air dan hanya mengirimkan sejumlah data yang sama yang terakhir dikirim oleh sistem *watermeter* sebelum *valve* tertutup.



**Gambar 17.** Kondisi valve Ketika tertutup

Berdasarkan hasil pengujian *feedback* sistem *watermeter* dapat membuka dan menutup *valve* menggunakan IoT akan sangat berguna bagi pihak PDAM. PDAM dapat dengan mudah melakukan pemutusan aliran air kepada pelanggan-pelanggan yang tidak membayar sejumlah harga yang harus dibayarkan. Maka dari itu meteran air yang telah disediakan oleh pihak PDAM harus disatukan dengan *valve* dan sistem perangkat keras yang telah di program.

## 4. Kesimpulan

Sistem *watermeter* telah berhasil dibangun dan memiliki tingkat akurasi yang tinggi. Dari hasil pengujian diperoleh rata-rata *error relative* dari sensor water flow sebesar 2,39%. Berdasarkan hasil pengujian pengiriman data volume air diperoleh sistem *watermeter* dapat bekerja secara *real time* dan dapat melakukan pemantauan penggunaan volume air dan menghitung harga yang harus dibayar pelanggan. Pembukaan dan penutupan *valve* bekerja sesuai dengan perintah pada *website* berdasar pada sistem yang telah ditetapkan. Sistem *watermeter* berbasis IoT telah bekerja dengan baik dan dapat melakukan umpan balik.

## Referensi

- [1] V. Gokilapriya and P. T. V. Bhuvaneshwari, "Design and development of SoC based residential water meter monitoring system," *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, vol. 2018-Janua, pp. 2163–2166, 2018, doi: 10.1109/WiSPNET.2017.8300143.
- [2] B. Sugiarto and Suharwanto, "Pengembangan Pemanfaatan Pengolahan Air Dalam Upaya Pemenuhan Kebutuhan Air di Dusun Temuireng , Desa Girisuko , Panggang , Gunungkidul Developing the Utilization of Water Treatment Technology to Fill Water Demand at Temuireng-Girisuko Village ," *Eksergi*, vol. 14, no. 2, pp. 40–52, 2017.
- [3] B. Sumantri and H. Parwiyanto, "Kualitas Pelayanan Perusahaan Daerah Air Minum (PDAM) Kabupaten Sragen," *J. Chem. Inf. Model.*, vol. 1, no. 1, pp. 11–24, 2017, doi: 10.1017/CBO9781107415324.004.
- [4] R. Wiryadinata and B. F. Butar-butur, "Rancang Bangun Alat Meteran Air Digital Menggunakan Sensor Aliran Air SEN-HZ21WA," *VOLT J. Ilm. Pendidik. Tek. Elektro*, vol. 3, no.

- 1, p. 26, 2018, doi: 10.30870/volt.v3i1.3585.
- [5] B. W. Dwi Putra Arief Rachman Hakim, Arief Budijanto, "Sistem Monitoring Penggunaan Air PDAM pada Rumah Tangga Menggunakan Mikrokontroler NODEMCU Berbasis Smartphone ANDROID," *J. IPTEK*, pp. 9–18, 2018, doi: 10.31284/j.iptek.2018.v22i2.
- [6] D. L. Yaddarabullah, "Perancangan Sistem Komunikasi Data Alat Pencatatan Meter," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. jaringan)*, vol. 3, no. 1, pp. 49–54, 2018.
- [7] D. Setiadi and M. Nurdin Abdul Muhaemin, "PENERAPAN INTERNET OF THINGS ( IoT ) PADA SISTEM MONITORING IRIGASI," *J. Infotronik*, vol. 3, no. 2, 2018.
- [8] A. Al-Naser and W. M. El-Medany, "A proposal for server-less cloud-based infrastructure for a smart metering system in the Kingdom of Bahrain," *2018 Int. Conf. Innov. Intell. Informatics, Comput. Technol. 3ICT 2018*, pp. 1–4, 2018, doi: 10.1109/3ICT.2018.8855729.
- [9] R. Koech, Y. Gyasi-Agyei, and T. Randall, "The evolution of urban water metering and conservation in Australia," *Flow Meas. Instrum.*, vol. 62, pp. 19–26, 2018, doi: 10.1016/j.flowmeasinst.2018.03.011.
- [10] M. Montginoul and A. Vestier, "Smart metering: A water-saving solution? Consider communication strategies and user perceptions first. Evidence from a French case study," *Environ. Model. Softw.*, vol. 104, pp. 188–198, 2018, doi: 10.1016/j.envsoft.2018.02.006.
- [11] A. Rahman and M. Suryanegara, "The development of IoT LoRa: A performance evaluation on LoS and Non-LoS environment at 915 MHz ISM frequency," *Proc. - Int. Conf. Signals Syst. ICSigSys 2017*, pp. 163–167, 2017, doi: 10.1109/ICSIGSYS.2017.7967033.
- [12] Y. Li, X. Yan, L. Zeng, and H. Wu, "Research on water meter reading system based on LoRa communication," *2017 IEEE Int. Conf. Smart Grid Smart Cities, ICSGSC 2017*, pp. 248–251, 2017, doi: 10.1109/ICSGSC.2017.8038585.